

Highly Expandable Reconfigurable Platform using Multi-FPGA based Boards

Saifullah Hammad

Department of Electrical & Electronics Engineering
Muhammad Ali Jinnah University (MAJU),
Islamabad, Pakistan

Muhammad Hasnain

Department of Electronics
Quaid-i-Azam University (QAU),
Islamabad, Pakistan

ABSTRACT

Reconfigurable computing has become an essential part of research since last few decades. By placing computationally intensive applications in the reconfigurable logic area of the system, the remarkable performance gains have been found. Among the different research directions in the domain of reconfigurable computing, the use of multiple reconfigurable devices has become the most promising solution for highly expandable reconfigurable computing platforms. In this research paper an emerging design of a Multi-FPGA based platform has been presented with the characteristics of being highly expandable for incorporating many devices according to future needs. The design of this platform is encountering the capability of being usable for highly scalable reconfigurable computing applications. The presented platform is based on an emerging concept of using multi-dimensional computing nodes with most optimal configuration propagation latencies.

Keywords

Reconfigurable computing, Multi-FPGA Systems, Multi-FPGA Boards, FPGA, Expandable Reconfigurable Platform.

1. INTRODUCTION

In recent years FPGA (Field Programmable Gate Array) based solutions have gained large attentions of the researchers, spatially reconfigurable systems are getting popularity day by day because of their flexible nature [5]. In past years FPGAs were used mostly for implementing digital or glue logic but now a day's their role have been increased exponentially fast beyond the implementing of digital or glue logic. The concepts of parallelism have gain much more attention to reduce the execution time generally required for intensive data processing. FPGAs can be treated as "Virtual Hardware" and such systems can be used for different types of applications, even the applications which require multidimensional computations. Normally these kinds of systems are developed under the naming umbrella of "Run-Time Reconfigurable or Dynamically Reconfigurable" Systems. Property of reconfigurability also provides the benefit of loading more configurations rather than sticking on single configuration for single hardware. Thus number of configurations depends upon the space available to hold them for the same hardware.

Researchers are trying to introduce more new and innovative ideas to use same hardware resource for multiple executable tasks. It means hardware must be capable of performing multiple tasks at the same time. Multiple FPGAs, depending upon the nature of application, are being used for the execution of different problems. Reconfigurable nature of

FPGAs make them more important to perform multiple problems. Reconfiguration of FPGA means reconfiguration of its logic blocks at gate level. Reconfigurability is the ability to dynamically allocation of resources to a computing device during run time. This property provides the benefit of reusing same hardware for other applications.

As one of the major characteristic of the reconfigurable computing is to allocate the resources of the computing device (FPGA in our case) during the run time for another task thus the reconfigurable time becomes a major problem when there is a frequent switch over between different modes of operations. There are several techniques available to overcome this problem. For example, reconfiguration of the portion of the chip according to the requirement [2] as is shown in the figure.1 below,



Figure 1: Reconfigurable memory map

Some other methods of reducing reconfiguration time are

1. By compressing the reconfiguration data
2. By introducing a configuration cache which in result will reduce the fetching time.

Flexibility of the reconfigurable hardware is also a very core issue and the point is how easily hardware resources are expandable to accommodate more logic. A large number of FPGA based systems have been developed until now for specific application based systems to multipurpose wide range systems. One or two FPGA based reconfigurable systems with some standard interfacing circuitry are commonly used and become more flexible by adding memories. The systems with hundreds of FPGAs are becoming the counterparts of super-computers for specific applications. In the FPGA based systems the most important thing is the topology used for FPGA interconnections. Two simple basic kind of FPGA interconnection topologies are

- 1 - Crossbar Topology
- 2 - Mesh Topology

The crossbar topology is simplest in designing. In crossbar topology there are two types of elements, one is called logic bearing FPGAs and other is called routing only FPGAs. Logic

bearing FPGAs contains the logic required by the system to be executed and the routing only FPGAs are responsible for data transferring among the FPGAs [3]. In this topology routing only FPGAs and logic bearing FPGAs are equal in number normally. In cross bar topology every routing FPGA is connected with every other logic bearing FPGA or vice versa as shown in fig.2.

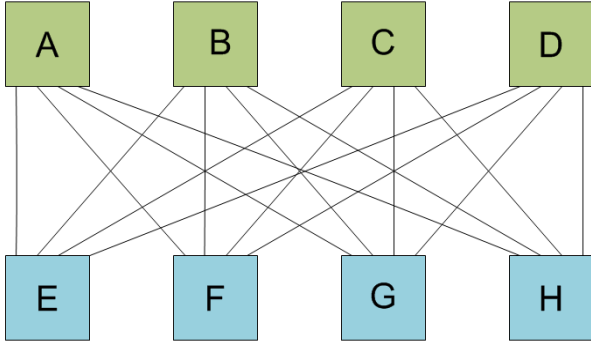


Figure 2: Crossbar topology

In the figure above FPGAs from A to D are routing only FPGAs and FPGAs from E to H are logic bearing FPGAs. All the logic resides in the logic bearing FPGAs. In this topology data routing between any two logic bearing FPGAs requires only one extra chip and that chip is routing only FPGA. Therefore delay in data transferring from one FPGA to any other FPGA is fixed and predictable because of the symmetry of the system. The major drawback of this topology is its expandability as its resources grows since every logic bearing FPGA is connected with every routing only FPGA therefore design will not be expandable and will be constrained to a specific size.

In mesh topology design FPGAs are connected with their nearest neighbors as shown in the figure.3.

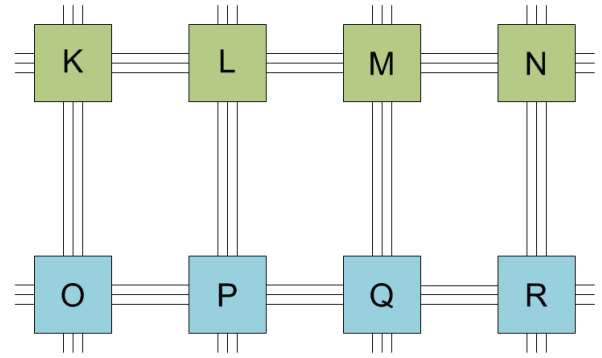


Figure 3: Mesh topology

The major advantage of this topology is its expandability as it can be expanded easily by adding resources on its edges. But there is also a disadvantage of inter chip delay and delay increases with the increase in size of the topology. Many 2D and 3D mesh topologies have been constructed until now [4].

There is another topology which is constructed on the basis of advantages of both Mesh and Crossbar topologies which is known as hierarchical Crossbar topology [5]. Besides these mentioned topologies several other topologies have also been introduced which are currently not under our scope.

There are many reconfigurable computing based research projects which emphasize on new and improved hardware designs for reconfigurable chips [6] [7] [8] [9][10][11][12][13][14]. In this paper we shall discuss a hardware solution which is not only easily reconfigurable but also easily expandable for extensive data computations and it consists of crossbar and mesh topologies.

2. HARDWARE ARCHITECTURE

Purposed reconfigurable hardware solution consists of a Base board and Daughter boards.

2.1 Base Board

Base board consists of two major processing elements which are FPGAs with dedicated RAMs (Random Access Memory) and a configurator responsible for configuring both the FPGAs according to the requirement as shown in the figure 4.

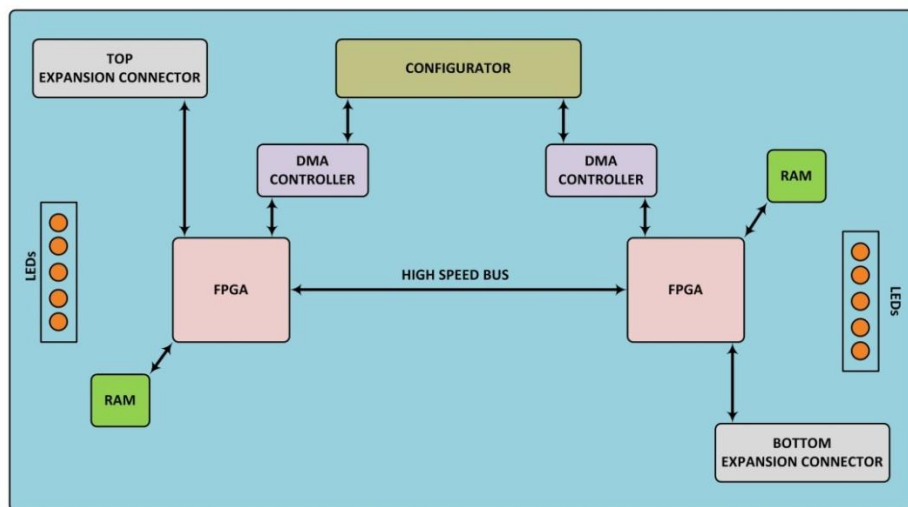


Figure 4: Base board

In multi FPGA based systems most common thing is memory which is normally connected to the FPGA directly. Normally the purpose of the memory is to provide temporary data storage and circuit emulation. Architecture shown in the Fig.4 has two dedicated RAMs for temporary data storage. Both FPGAs are connected directly with each other through a high speed bus for fast transferring of data. There are two dedicated DMA (Direct Memory Access) controllers [15] for both the FPGAs. DMA controllers play a vital role when there is a frequent switch over between configurations. Due to DMA controller data fetching time reduces significantly while FPGA keeps processing data without disturbance. Configurator on the base board is a removable device. Configurator consists of ROMs (Read Only Memory) and a controller as shown in the figure 5. ROMs are configuring logic bearing devices. ROMs in the configurator are flexible in terms of quantity. If more configuring logic is required then more ROMs can be placed on the configurator and this dependency is purely application specific.

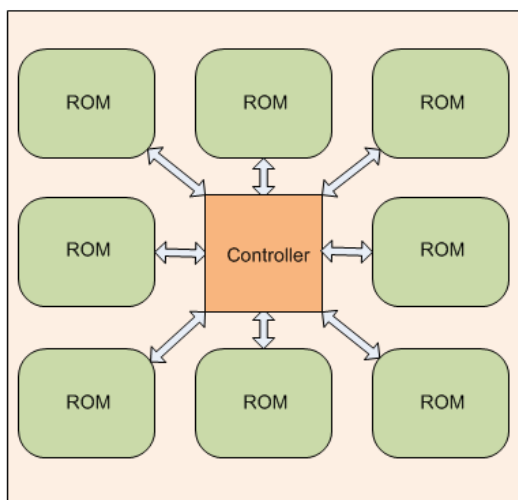


Figure 5: Configurator

Therefore to avoid wastage of resources configurator is a removable device. Task of controller in the configurator is to control configuring and reconfiguring operations of FPGAs through ROMs. There are total ten (10) status LEDs (Light Emitting Diodes) on the two edges of the base board. Five LEDs are placed on one edge and remaining five on the opposite edge of the board as shown in the figure 4. These LEDs can be used for different purposes. For example these LEDs can be used for hardware/software debugging or can be used for showing different statuses of running applications. There are two expansion slots on the base board, one is top expansion slot and second is bottom expansion slot. These slots are placed for daughter board attachments.

Base board has two independent roles. If only the base board is used and no daughter cards are attached on the expansion slots than this is a simple dual FPGA based system in which both the FPGAs are the main processing elements connected through a high speed bus for data transferring.

Base board plays its second role when daughter boards are attached with it through its expansion slots. If one or two daughter boards are attached with it than FPGAs placed on the Base board act like a temporary cache for data storage of daughter boards FPGAs and also used as routing only FPGAs between two daughter boards. Please note that Base board and

daughter boards should be stacked in such a way that Base board must be in the middle of the stack as shown in figure 6.

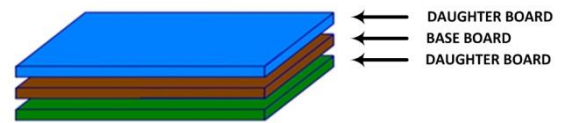


Figure 6: Board Stacking Architecture

Stacking the boards according to the figure6 provides the benefit of predictable delay as there are only two routing only FPGAs (Base board FPGAs) between the daughter boards. By adding more daughter boards in such a way predictable delay remains the same as number of routing only chips remains constant which is two in our case. Thus the communication between any two FPGAs of any two top and bottom daughter boards are only through two routing only FPGAs. We achieve another benefit of such stacking in the form of clock skewing in fast speed processing applications if a single clock is used for the entire design. For example, suppose that the entire design is driven through a single clock and clock is being produced by the Base board. Now we place the daughter boards only on the single side of the Base board and when the design will grow than clock skewing will become a critical issue which in return will compromise the expandability of the design.

Communication between all the boards is purely packet based with unique headers of each FPGA placed on the boards. Packet based communication provides one benefit and one drawback. Benefit is that it does not compromises the expandability as IOs of FPGAs are not dedicated for daughter boards or any individual FPGAs and drawback is that computational overheads increases with increase of daughter boards and logic also becomes more complex.

2.2 Daughter Board for Parallel Processing Applications

In daughter board there are four FPGAs connected to each other through crossbar topology. Crossbar topology is modified in such a way that there are no routing only FPGAs on the board. FPGAs are configured in such a way that a small portion of resource in each FPGA is dedicated for inter FPGA routing. This technique is called "Local run-time Reconfiguration" [16]. In this technique different algorithms can be loaded on different portions of the reconfigurable chip. In this way different mappings can coexists at the same time on single reprogrammable chip. Thus reprogrammable chip can be called as hardware cache. Usage of multiple mappings at the same time on the single chip also provides the benefit of less reprogrammable time. As it is quite time worthy to reprogram a small portion of the chip instead of the complete chip.

There are only four (04) FPGAs on each daughter board so there is no need to place extra chips for only inter chip routing and this task can be performed perfectly by dedicating a small portion of resources in each FPGA. In such a way each FPGA on the daughter board has become not only logic bearing FPGA but also a routing only FPGA. Each FPGA is connected directly with remaining three (03) FPGAs and thus inter chip delay of data transferring is not only minimum but also predictable.

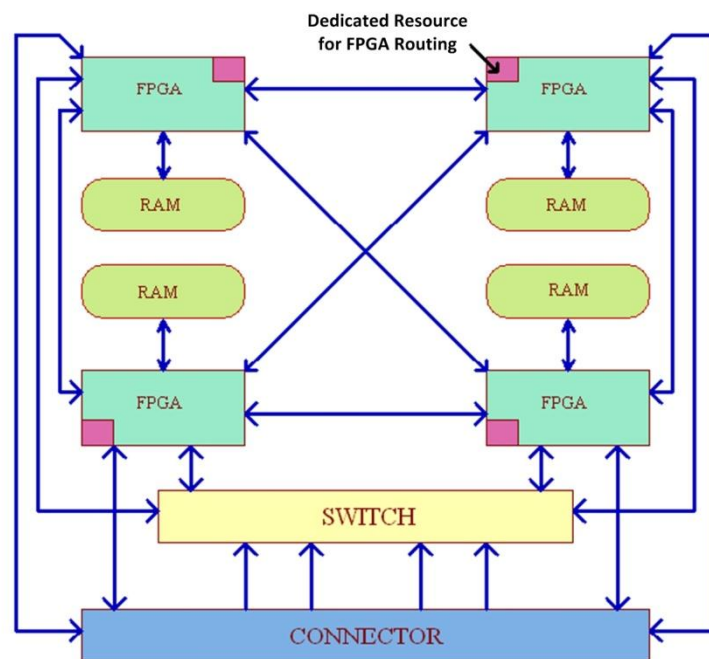


Figure 7: Daughter board for parallel computing applications

Each FPGA on the daughter card has its own dedicated RAM for fast data processing applications as shown in figure 7.

Connector/Expansion slot placed on the daughter board provides the connectivity on both top and bottom sides for expandability purposes. By using this connector we can stack more boards on each other which provide easy expandability. Each FPGA of daughter board is also directly connected to the

main board FPGAs through this connector. Switch placed on the daughter board is used to configure the FPGAs of the daughter board through main board. These FPGAs can be configured simultaneously for the same logic or individually for multiple logics through this switch. Functionality of the switch is more elaborated in the figure 8 and the table shown below.

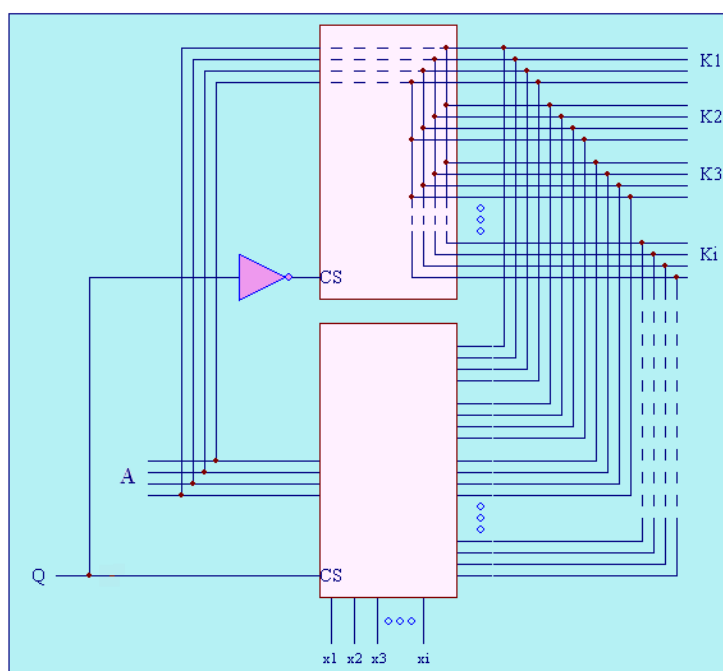


Figure 8: Reconfigurable Switch

Chip Select	Selection	Output	Result
Q = 0	$x_1=x_2=x_3\ldots=x_i=X$	$A=K_1=K_2=K_3\ldots=K_i$	Simultaneous configuration of all FPGAs
Q = 1	$x_1=1, x_2=x_3\ldots=x_i=0$ $x_2=1, x_1=x_3\ldots=x_i=0$ $x_3=1, x_1=x_2\ldots=x_i=0$ \vdots \vdots $x_i=1, x_1=x_2\ldots=x_{i-1}=0$	$A=K_1, K_2=K_3\ldots=K_i=0$ $A=K_2, K_1=K_3\ldots=K_i=0$ $A=K_3, K_1=K_2\ldots=K_i=0$ \vdots \vdots $A=K_i, K_1=K_2\ldots=K_{i-1}=0$	Individual configuration of FPGAs

Table: Switch Configurability Modes

Output of all the FPGAs are directly connected to the expansion connector which in result saves the inter chip delay and also provides the predictability about the time delay. Time delay of interconnects can be calculated by the following formula.

$$TD = \frac{x \sqrt{\epsilon}}{c} \quad [17]$$

Where

TD = Time Delay

x = Length of the trace (interconnect) in meters

ϵ = Dielectric Constant

c = Speed of Light in vacuum

2.3 Daughter Board for Arithmetic and Floating Point Operations

Some applications required extensive arithmetic and floating point calculations. Therefore sometimes we need out of the box solutions as FPGAs are not good enough for these types of applications. The figure 9 below provides us the flexibility for these kinds of applications.

In this daughter board as shown in figure 9, there are four (04) FPGAs connected with each other directly to avoid inter chip delays. Each FPGA has its dedicated resource for inter FPGA routing. All four FPGAs have their dedicated ALUs (Arithmetic Logic Units) and FPUs (Floating Point Units). Thus with the presence of these dedicated resources each FPGA can perform arithmetic and floating point operations very rapidly. This solution is very efficient for such kind of applications in which independent arithmetic and floating point operations are required. There are two RAMs on the design shared with each couple of FPGAs as shown in the figure 9. Presence of RAMs provides temporary storage of data and thus as result fast processing. Switch placed on the board is used to configure FPGAs as elaborated in figure 8 and table above. Connector in the design is used for expansion and we can plug more such boards on each other.

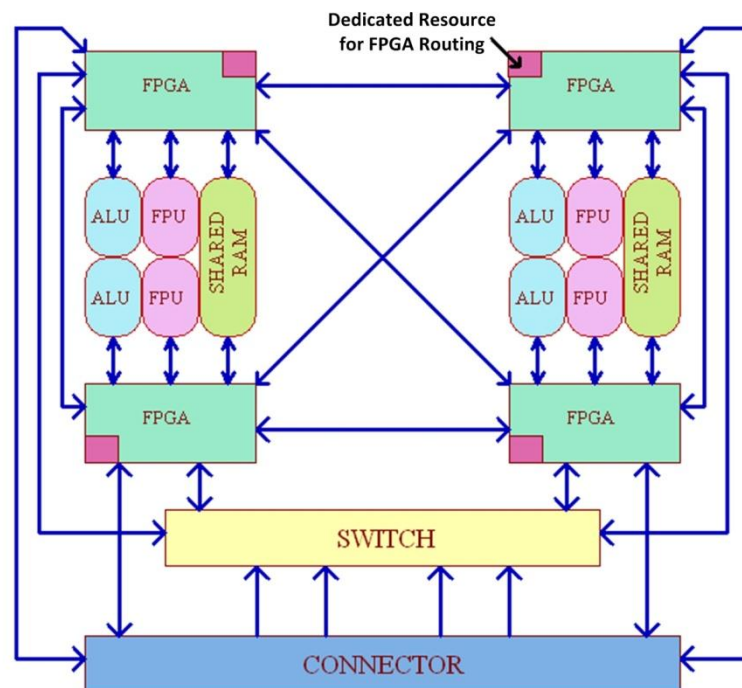


Figure 9: Daughter board for arithmetic and floating point operations

3. CONCLUSION

Multi-FPGA boards are designed to support highly scalable computing platforms. The multiple computing nodes are placed at different dimensions in such a way that the time required to move data from device to device according to the application requirements, should be minimum. The cross-bar switching technology has been used in the design of platform so that each computing node (FPGA) can be peer-to-peer connected with other devices. The dedicated communication paths have been provided for streaming the configuration data among the devices and hence platform is negatively hitting the optimal cost model. Since the multi-dimensional structure was requirement for making the platform highly expandable, hence multi-layers PCB design concepts have been used. The presented platform is expected to be an effective design for emerging expandable FPGA based computing systems.

4. REFERENCES

- [1]. Scott Hauck, "The Role of FPGAs in Reprogrammable Syatems" in: Proceedings of the IEEE, Vol.86, No.4, pp. 615-639, April, 1998.
- [2]. k.Compton, S. Hauck, Reconfigurable Computing: a survey of systems and software, ACM Computing Surveys 34 (2) (2002).
- [3]. P. K. Chan, M. Schlag, M. Martin, "BORG: A Reconfigurable Prototyping Board Using Field-Programmable Gate Arrays", Proceedings of the 1st International ACM/SIGDA Workshop on Field-Programmable Gate Arrays, pp. 47-51, 1992.
- [4]. K. Yamada, H. Nakada, A. Tsutsui, N. Ohta, "High-Speed Emulation of Communication Circuits on a Multiple-FPGA System", 2nd International ACM/SIGDA Workshop on Field-Programmable Gate Arrays, 1994.
- [5]. J. Varghese, M. Butts, J. Batcheller, "An Efficient Logic Emulation System", IEEE Transactions on VLSI Systems, Vol. 1, No. 2, pp. 171-174, June 1993.
- [6]. Lu Wan, Chen Dong and Deming Chen "A Coarse-Grained Reconfigurable Architecture with Compilation for High Performance", International Journal of Reconfigurable Computing Volume 2012 (2012), Article ID 163542
- [7]. Rajeev Wankar and Rajendra Akerkar "Reconfigurable architectures and algorithms: A research survey (2009)" International Journal of Computer Science and Applications (2009), Techno mathematics Research Foundation Vol. 6, No. 1, pp. 108 – 123
- [8]. Mateusz Majer, Jürgen Teich, Ali Ahmadinia, and Christophe Bobda "The Erlangen Slot Machine: A Dynamically Reconfigurable FPGA-Based Computer", in Journal of VLSI Signal Processing Systems, Kluwer Academic Publisher (Springer), Vol. 47, No. 1, pp. 15-31, April 2007.
- [9]. Lu Wan, Chen Dong, and Deming Chen "A New Coarse-Grained Reconfigurable Architecture with Fast Data Relay and Its Compilation Flow", Symposium on Application Accelerators in High-Performance Computing (SAAHPC), July 2009.
- [10]. C. Bolchini, L. Fossati, D. Merodio Codinachs, A. Miele, C. Sandionigi "A Reliable Reconfiguration Controller for Fault-Tolerant Embedded Systems on Multi-FPGA Platforms", Defect and Fault Tolerance (DFT) Proceedings of the 2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems Pages 191-199
- [11]. RAW Project, Laboratory for Computer Science, MIT. <http://cag-www.lcs.mit.edu/raw/>
- [12]. Garp Project, BRASS Research Group. UC Berkeley <http://brass.cs.berkeley.edu/garp.html>
- [13]. PipeRench Project, Carnegie Mellon University <http://www.ece.cmu.edu/research/piperench>
- [14]. Andreas Weisensee, Darran Nathan: Project Proteus, "A Self-Reconfigurable Computing Platform Hardware Architecture" arXiv:cs/0411075v1 [cs.AR] 20 Nov 2004.
- [15]. G. Venkataramani, W. Najjar, F. Kurdhai, N. Bagherzadeh, W. Bohm: A compiler framework of mapping applications to a coarse-grained reconfigurable computer architecture, in: Proceedings of the 2001 international conference on compilers, Architecture and Synthesis for Embedded Systems (CASE'01) 2001, pp. 116-125
- [16]. P. Lysaght, "Dynamically Reconfigurable Logic in Undergraduate Projects", in W. Moore, W. Luk, Eds., FPGAs, Abingdon, England: Abingdon EE&CS Books, pp. 424-436, 1991.
- [17]. Stephen H. Hall, Garrett W. Hall, James A. McCall, "A Handbook of Interconnect Theory and Design Practices"